Railroad Signaling Block Design Tool

Kenneth Truex – [ktruex2012@my.fit.edu](mailto:ktruex2012@my.fit.edu)

Zachary McHenry – [zmchenry2011@my.fit.edu](mailto:zmchenry2011@my.fit.edu)

Christopher Diebold – [cdiebold2012@my.fit.edu](mailto:cdiebold2012@my.fit.edu)

Chad Mason – [cmason2011@my.fit.edu](mailto:cmason2011@my.fit.edu)

**Faculty Sponsor**

Phillip Bernhard – [pbernhar@fit.edu](mailto:pbernhar@fit.edu)

**Progress of current milestone**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Task | Completion % | Kenneth Truex | Christopher Diebold | Chad Mason | Zachary McHenry | To Do |
| Create the user forms | 70% |  |  | 70% |  | Implement GE’s UI Template |
| Finish the algorithms | 100% | 100% |  |  |  |  |
| Query the database | 100% |  | 100% |  |  |  |
| Look into multithreading | 85% |  |  | 30% |  | Implement Semaphores |
| Implement Excel Parser | 100% |  |  |  | 100% |  |

**Summary of Accomplished Tasks for Milestone 3**

Create the user forms: Currently, we are working on taking our original demo UI design and altering it to resemble the UI design template provided to us by GE. Upon initial review, our demo design was a very close match to what GE had in its examples. We are 70% done with the alterations of the portions that were not a match.

Finish the algorithms: For Phase one of this project, GE has instructed us to only implement the Safe Breaking Distance. The algorithm designed to perform this calculation has been implemented and accepted by GE. It is currently used to calculate the SBD of a given track segment and store it in the corresponding Track Segment object.

Query the database: Upon initial launch, the user can select to load file from Excel. The program then sends the Excel file to the Excel Parser that we created. The relevant information is extracted then stored, for further use, in the database using the newly implemented, insert statements.

Look into multithreading: We currently have the Excel parsing occurring in a separate thread so as not to stop the entire program while waiting for the parser to finish. We are 85% complete because we still need to implement a Semaphore in order to prevent a user from trying to access the data before it is finished being parsed.

Implement Excel Parser: For Phase one of the project, GE instructed us to only concern ourselves with receiving an Excel file as input to the tool. What we agreed upon was that the Excel file would be sent to an implemented parser that extracts the relevant information and stores it directly in the database.

**Summary of Contribution of Each Team Member**

Ken: For this milestone, I completed the Safe Braking Distance Algorithm. I ensured that when my function is called from the Track Segment class, it returns the proper value to be stored in the Track Layout object. I also added assert statements to ensure the function never returns a negative value (That would be impossible).

Chris: For this milestone, I completed the functionality for basic backend functionality. Specifically, the insert, delete, update, and find functions. The most challenging of these functions was the find function since on some of the track segments there are some values that are many to one. Meaning given a track segment object there is some value for say worst case grade for example that has more than one value associated with it. The solution was to store all the multi values in a list.

Chad: For this milestone, I created more of the forms needed in the program. Not all of them could be finished because we are still waiting for updated GUI requirements from GE. Began to implement some multithreading so the program doesn’t come to a complete halt when we are importing an excel file and putting it into the database.

Zach: For this milestone, I created a set of classes that extract data from a Microsoft Excel file. The intention of these classes, and the functions that accompany them is to parse the valuable data from a Microsoft Excel file and use this data for various calculations, which, before now, were done by hand. The data that is extracted will be put into a database and then used to calculate safe breaking distance as of right now. Parsing a Microsoft Excel file will allow GE to store all of the data that is pertinent to a railroad track in an Excel file, in the same way that GE is accustomed to.

**Summary of Milestone 3 Tasks**

Create the user forms: It took GE a significant amount of time to come up with an official document displaying their expectations for the UI. Fortunately, once we did receive the documentation, which was on April 14, 2014, we found that it was not too different from the demo UI that we had created.

Finish the algorithms: Initial thoughts from the team were that only implementing Safe Breaking Distance in Phase one as opposed to at least one or two other calculations was not optimal. GE reassured us that the other four calculations are rather simple and easy to implement. For Phase one, they want a working, demonstrable product that can be field tested on engineers who would then provide feedback.

Query the database: Implemented and tested the insert, remove, update, and find functions for the database. To test these functions I did some queries in MySQL and then did the same queries in the code and then manually verified the results. I did this for ten different track segments. Also, when testing the find function I tested valid and invalid values ensuring that the proper exception handling was in place. At this point database work is 90% finished based on the requirements of the project. The remaining work to be done is just overloading the insert, remove, update, and find functions for the other objects that the user has access too.

Look into multithreading: After initial testing of the Excel Parser, we realized that it took approximately 5-10 seconds to parse all relevant information out of a given sample file. We did not want this to hold the user up from conducting other tasks not pertinent to the information being parsed. We decided to create a thread that the parser would run in thus allowing the user to not have to wait. We are currently working on a semaphore implementation in order to ensure that the user does not try to access the unparsed data.

Implement Excel Parser: GE informed us that they want the information to be parsed from a Microsoft Excel formatted file. They also informed us that, for right now, we can assume that the file layout will always be the same. We developed an Excel Parser that will take a file as input and pull out all relevant information and store it both in the database and create a Track Segment object out of it.

**Plan for Milestone 4**

* Finish altering UI to match GE’s design template
* Broad testing to ensure basic functionality
* Prepare for Phase 1 product demonstration to GE Executives
* Sit down with GE and plan out summer goals in order to prepare for Phase 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Kenneth Truex | Christopher Diebold | Chad Mason | Zachary McHenry |
| Finish UI |  |  | 75% |  |
| Testing |  | 75% |  | 75% |
| Prepare for Demo | 75% |  |  |  |
| Summer Goals | 25% | 25% | 25% | 25% |

**Sponsor Feedback on Each Task for the Current Milestone**

Create the user forms:

Finish the algorithms:

Query the database:

Look into multithreading:

Implement Excel Parser:

Sponsor Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Sponsor Evaluation**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kenneth Truex | 0 | 1 | 2 | 3 | 4 | 5 | 5.5 | 6 | 6.5 | 7 | 7.5 | 8 | 8.5 | 9 | 9.5 | 10 |
| Christopher Diebold | 0 | 1 | 2 | 3 | 4 | 5 | 5.5 | 6 | 6.5 | 7 | 7.5 | 8 | 8.5 | 9 | 9.5 | 10 |
| Zachary McHenry | 0 | 1 | 2 | 3 | 4 | 5 | 5.5 | 6 | 6.5 | 7 | 7.5 | 8 | 8.5 | 9 | 9.5 | 10 |
| Chad Mason | 0 | 1 | 2 | 3 | 4 | 5 | 5.5 | 6 | 6.5 | 7 | 7.5 | 8 | 8.5 | 9 | 9.5 | 10 |

Sponsor Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_